ATTORNEY DOCKET NO.
AUS920030139US1 (IBM 2638000)

PATENT APPLICATION
SERIAL NO. 10/645,024

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

In re application of: Kahle et al.	§	
	§	
Serial Number: 10/645,024	§	
	§	Group Art Unit: 2115
Filed: August 21, 2003	§	
	§	
For: POWER THROTTLING METHOD AND APPARATUS	§	Examiner: Dennis Butler
	§	

AFFIDAVIT OF
DAVID J. SHIPPY
UNDER RULE 131

I, David J. Shippy, first being duly sworn, do hereby state that:

1. I am over the age of 21 years, am of sound mind, am one of the joint inventors named in the above-identified patent application (the "Application") and have personal knowledge of the facts stated in this affidavit.

2. Prior to December 18, 2002, the subject matter claimed by each of Claims 16 through 36 of the Application, as provided in response to the Office Action for the Application dated August 21, 2003, was actually reduced to practice in the United States. Such Claims are referred to within this affidavit as the "Pending Claims" and are attached as Exhibit A.

3. I created an internal IBM Disclosure document subsequent to reduction to practice of the subject matter in the Pending Claims, and also prior to December 18, 2002. The "Main Idea" described in the IBM Disclosure document and detailing the subject matter of the Pending Claims is attached as Exhibit B.

4. As shown, Exhibit B clearly demonstrates conception of each and every claimed element in the Pending Claims of Exhibit A. A mapping of limitations from the independent Pending Claims of Exhibit A to the disclosure of Exhibit B is attached hereto as Exhibit C.

Further affiant sayeth not.

David J. Shippy

ATTORNEY DOCKET NO.
AUS920030139US1 (IBM 2638000)

PATENT APPLICATION
SERIAL NO. 10/645,024

STATE OF TEXAS)
) SS.
COUNTY OF TRAVIS)

Sworn and subscribed before me this 2nd day of April, 2007, by David J. Shippy.

Notary Public

(Seal)

LAURA SUSAN LINDSEY
Notary Public
STATE OF TEXAS
My Comm. Exp. 12-05-09

Exhibit "A" (Pending Claims)

16. (Previously Presented) A method of conserving power in a computer processor, comprising:

reading a software-accessible control register;

determining an idle status of a subunit of the computer processor based on the control register;

providing a clock signal to the subunit based on the determined idle status; and

providing a power voltage to the subunit based on the determined idle status.

17. (Previously Presented) The method as recited in Claim 16, wherein the control register comprises a plurality of architected bits in a machine state register and at least one bit of the plurality of architected bits is associated with at least one subunit of the computer processor.

18. (Previously Presented) The method as recited in Claim 17, wherein determining the idle status comprises reading the at least one bit associated with the at least one subunit of the computer processor.

19. (Previously Presented) The method as recited in Claim 16, further comprising setting one or more of a plurality of bits in the control register based on an idle status of a subunit of the computer processor.

20. (Previously Presented) The method as recited in Claim 16, wherein the computer processor comprises data flow circuitry comprising a plurality of data flow sections, at least one data flow section configured as a subunit of the computer processor.

21. (Previously Presented) The method as recited in Claim 16, wherein the computer processor comprises upper and lower bit data register circuitry portions, at least the upper bit data register circuitry portion configured as a subunit of the computer processor.

22. (Previously Presented) The method as recited in Claim 16, wherein:

the computer processor comprises partitioned dataflow registers comprising a lower portion register consistent in size with the lowest instruction width software to be used in the computer processor; and

the control register comprises an architected control register bit indicating the width of the greatest instruction width presently being used by the computer processor.

23. (Previously Presented) The method as recited in Claim 16, wherein the computer processor comprises partitioned arithmetic logic units (ALUs), comprising an upper ALU and a lower ALU, at least the upper ALU configured as a subunit of the computer processor.

24. (Previously Presented) The method as recited in Claim 16, wherein the computer processor comprises a floating point logic unit (FPU), the FPU configured as a subunit of the computer processor.

25. (Previously Presented) An apparatus for conserving power in a computer processor, comprising:

a software accessible control register having predetermined bit positions indicating subunits of the computer processor;

a local clock buffer coupled to the control register and configured to provide a clock signal to a subunit based on the predetermined bit position associated with the subunit; and

a voltage signal coupled to the control register and configured to provide a power voltage to the subunit based on the predetermined bit position associated with the subunit.

26. (Previously Presented) The apparatus as recited in Claim 25, wherein the control register comprises a plurality of architected bits in a machine state register and at least one bit of the plurality of architected bits is associated with at least one subunit of the computer processor.

27. (Previously Presented) The apparatus as recited in Claim 25, further comprising software configured to set one or more of the predetermined bit positions based on an idle status of a subunit of the computer processor.

28. (Previously Presented) The apparatus as recited in Claim 25, wherein the computer processor comprises data flow circuitry comprising a plurality of data flow sections, at least one data flow section configured as a subunit of the computer processor.

29. (Previously Presented) The apparatus as recited in Claim 25, wherein the computer processor comprises upper and lower bit data register circuitry portions, at least the upper bit data register circuitry portion configured as a subunit of the computer processor.

30. (Previously Presented) The apparatus as recited in Claim 25, wherein:

the computer processor comprises partitioned dataflow registers comprising a lower portion register consistent in size with the lowest instruction width software to be used in the computer processor; and

the control register comprises an architected control register bit indicating the width of the greatest instruction width presently being used by the computer processor.

31. (Previously Presented) The apparatus as recited in Claim 25, wherein the computer processor comprises partitioned arithmetic logic units (ALUs), comprising an upper ALU and a lower ALU, at least the upper ALU configured as a subunit of the computer processor.

32. (Previously Presented) The apparatus as recited in Claim 25, wherein the computer processor comprises a floating point logic unit (FPU), the FPU configured as a subunit of the computer processor.

33. (Previously Presented) A computer program product for conserving power in a computer processor, the computer program product having a medium with a computer program embodied thereon, the computer program comprising:

computer program code for reading a software-accessible control register;

computer program code for determining an idle status of a subunit of the computer processor based on the control register;

computer program code for providing a clock signal to the subunit based on the determined idle status; and

computer program code for providing a power voltage to the subunit based on the determined idle status.

34. (Previously Presented) The computer program product as recited in Claim 33, wherein the control register comprises a plurality of architected bits in a machine state register and at least one bit of the plurality of architected bits is associated with at least one subunit of the computer processor.

35. (Previously Presented) The computer program product as recited in Claim 34, wherein computer program code for determining the idle status comprises computer program code for reading the at least one bit associated with the at least one subunit of the computer processor.

36. (Previously Presented) The computer program product as recited in Claim 33, further comprising computer program code for setting one or more of a plurality of bits in the control register based on an idle status of a subunit of the computer processor.

Main Idea for Disclosure AUS8-2002-0981

Prepared for and/or by an IBM Attorney - IBM Confidential

[redacted]

Title of disclosure (in English)
Power Throttling using Architected State

Main Idea of disclosure
1. Describe your invention, stating the problem solved (if appropriate), and indicating the advantages of using the invention.

Power MSR.pdf

2. How does the invention solve the problem or achieve an advantage,(a description of "the invention", including figures inline as appropriate)?
See attachment above.

3. If the same advantage or problem has been identified by others (inside/outside IBM), how have those others solved it and does your solution differ and why is it better?

4. If the invention is implemented in a product or prototype, include technical details, purpose, disclosure details to others and the date of that implementation.

[redacted]

Exhibit B

Title: Power throttling using Architected State [redacted]

Inventors: David Shippy, Jim Van Norstrand, Jim Kahle

1. Describe your invention, stating the problem solved (if appropriate), and indicating the advantages of using the invention.

Most processor architectures have an architected (software accessible) control register which controls various aspects of the design. In the PowerPC architecture there is a Machine State Register (MSR) which provides this function. This architected register controls such things as 64 bit mode, Hypervisor state, Floating-Point Available, etc.

As processor architectures evolve, the width of the instruction set architecture has increased over time. Original Central Processing Units (CPU) were 8 bits. Over time the width has been increased to up to 64 bits. However, as the width increases, there remains alot of software which is written for a previous generation's smaller width. Many of today's CPUs are 64 bit, but much of the code is still 32 bits. In the PowerPC architecture it is under software control to switch between 32 bit and 64 bit mode via a bit in the MSR register.

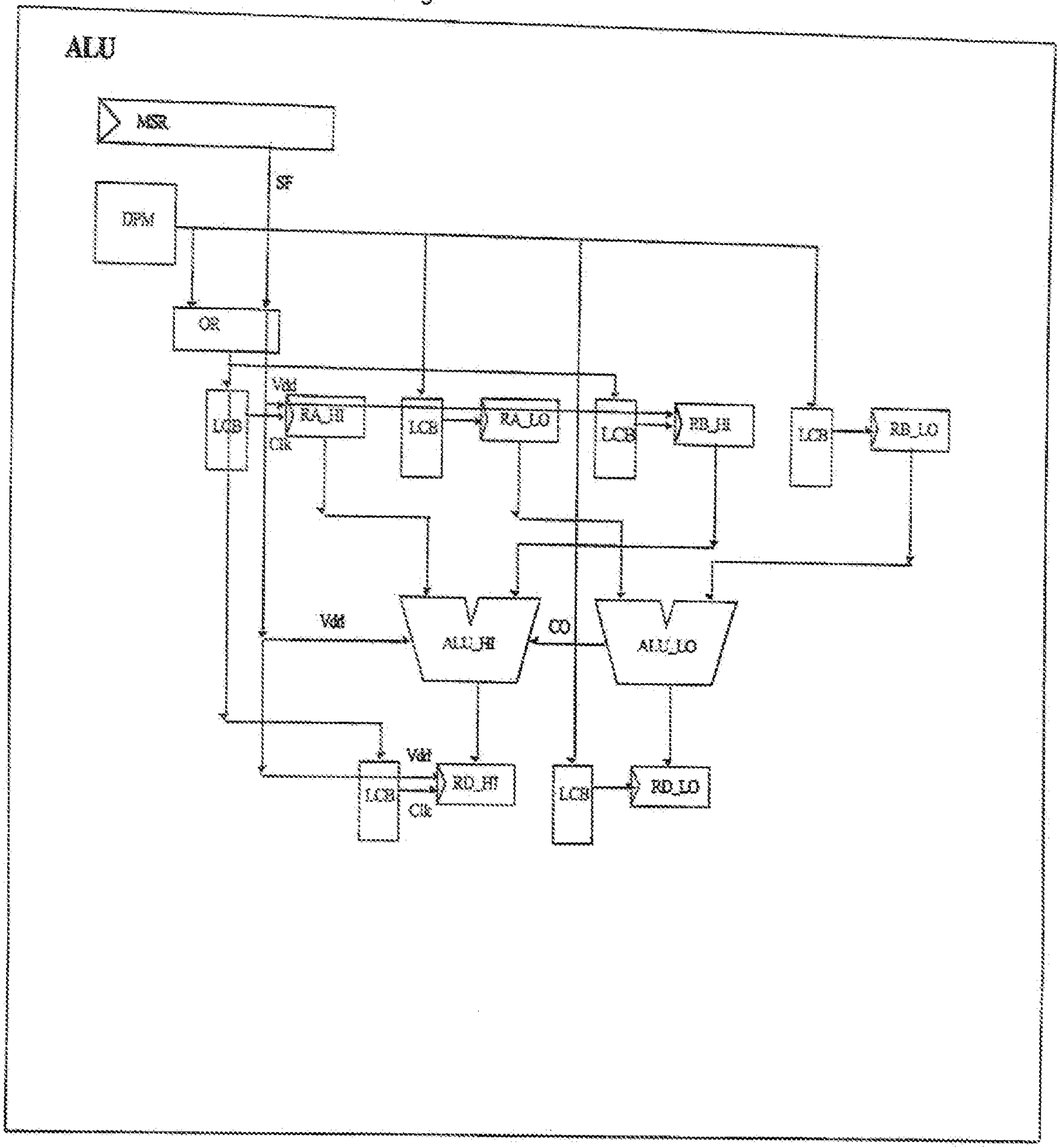
This invention provides a mechanism to use architected bits in the MSR to reduce power in the processor. For example, when the CPU is put in 32 bit mode, the hardware shuts off clocks for all of the upper 32 bits of the dataflow. In addition it shuts down the power supply for all of the upper 32 bits of register and dataflow logic. Another example is using the Floating-Point Available (FP) bit in the MSR to shutdown power and clocks to the entire floating point unit (FPU).

2. How does the invention solve the problem or achieve an advantage,(a description of "the invention", including figures inline as appropriate)?

Figure 1 shows a high level block diagram of the invention. The MSR(SF) bit feeds logic which shuts down the clocks for the upper half of all registers which transfer data. In addition it activates the power supply signal (Vdd) for the registers and logic for the upper 32 bits of dataflow. As seen in the figure, there is a Local Clock Buffer (LCB) for the lower 32 bits and upper 32 bits of each dataflow register. There is logic to shutdown the LCB's for normal dynamic power management which is contained in the DPM block. For the upper 32 bits of each register, the MSR(SF) is logically ORed with the DPM shutoff signal to turn clocks (Clk) off coming out of the LCBs. In this example a 64 bit ALU function is divided into the lower 32 bits (ALU_LO) and the upper 32 bits (ALU_HI). In 64 bit mode the MSR(SF) bit is a logical '1'b. In this mode all registers and dataflow are active and the carry-out of the lower ALU (CO) propagates to the upper ALU to form an entire 64 bit result. In 32 bit mode the MSR(SF) bit is a logical '0'b. In this mode the upper registers and dataflow macros are not clocked and they do not receive power (Vdd=0).

The MSR(FP) bit would be used in a similar manner to shutoff all clocks and power to all macros and registers in the FPU.

Figure1.ALU with Clock and Power Gating



[redacted]

Exhibit "C" (Annotated Claims)

Below are Claims 16-36 of Exhibit A, wherein the independent Claims 16, 25, and 33 are annotated to reflect where in Exhibit B their support can be found. Applicants have not provided every single instance of support, but rather an exemplary support citation for each limitation of the annotated Claims.

16. (Previously Presented) A method of conserving power in a computer processor, comprising:

reading a software-accessible control register; {page 1, para. 1, lines 1-2}

determining an idle status of a subunit of the computer processor based on the control register; {page 1, para. 2, lines 4-10}

providing a clock signal to the subunit based on the determined idle status; and {page 1, para. 2, lines 9-10; page 2}

providing a power voltage to the subunit based on the determined idle status. {page 1, para. 2, lines 9-10; page 2}

17. (Previously Presented) The method as recited in Claim 16, wherein the control register comprises a plurality of architected bits in a machine state register and at least one bit of the plurality of architected bits is associated with at least one subunit of the computer processor.

18. (Previously Presented) The method as recited in Claim 17, wherein determining the idle status comprises reading the at least one bit associated with the at least one subunit of the computer processor.

19. (Previously Presented) The method as recited in Claim 16, further comprising setting one or more of a plurality of bits in the control register based on an idle status of a subunit of the computer processor.

20. (Previously Presented) The method as recited in Claim 16, wherein the computer processor comprises data flow circuitry comprising a plurality of data flow sections, at least one data flow section configured as a subunit of the computer processor.

21. (Previously Presented) The method as recited in Claim 16, wherein the computer processor comprises upper and lower bit data register circuitry portions, at least the upper bit data register circuitry portion configured as a subunit of the computer processor.

22. (Previously Presented) The method as recited in Claim 16, wherein:

the computer processor comprises partitioned dataflow registers comprising a lower portion register consistent in size with the lowest instruction width software to be used in the computer processor; and

the control register comprises an architected control register bit indicating the width of the greatest instruction width presently being used by the computer processor.

23. (Previously Presented) The method as recited in Claim 16, wherein the computer processor comprises partitioned arithmetic logic units (ALUs), comprising an upper ALU and a lower ALU, at least the upper ALU configured as a subunit of the computer processor.

24. (Previously Presented) The method as recited in Claim 16, wherein the computer processor comprises a floating point logic unit (FPU), the FPU configured as a subunit of the computer processor.

25. (Previously Presented) An apparatus for conserving power in a computer processor, comprising:

a software accessible control register having predetermined bit positions indicating subunits of the computer processor; {page 1, para. 1, lines 1-2; para. 2, lines 13-14}

a local clock buffer coupled to the control register and configured to provide a clock signal to a subunit based on the predetermined bit position associated with the subunit; and {page 1, para. 2, lines 1-10; page 2}

a voltage signal coupled to the control register and configured to provide a power voltage to the subunit based on the predetermined bit position associated with the subunit. {page 1, para. 2, lines 1-10; page 2}

26. (Previously Presented) The apparatus as recited in Claim 25, wherein the control register comprises a plurality of architected bits in a machine state register and at least one bit of the plurality of architected bits is associated with at least one subunit of the computer processor.

27. (Previously Presented) The apparatus as recited in Claim 25, further comprising software configured to set one or more of the predetermined bit positions based on an idle status of a subunit of the computer processor.

28. (Previously Presented) The apparatus as recited in Claim 25, wherein the computer processor comprises data flow circuitry comprising a plurality of data flow sections, at least one data flow section configured as a subunit of the computer processor.

29. (Previously Presented) The apparatus as recited in Claim 25, wherein the computer processor comprises upper and lower bit data register circuitry portions, at least the upper bit data register circuitry portion configured as a subunit of the computer processor.

30. (Previously Presented) The apparatus as recited in Claim 25, wherein:

the computer processor comprises partitioned dataflow registers comprising a lower portion register consistent in size with the lowest instruction width software to be used in the computer processor; and

the control register comprises an architected control register bit indicating the width of the greatest instruction width presently being used by the computer processor.

31. (Previously Presented) The apparatus as recited in Claim 25, wherein the computer processor comprises partitioned arithmetic logic units (ALUs), comprising an upper ALU and a lower ALU, at least the upper ALU configured as a subunit of the computer processor.

32. (Previously Presented) The apparatus as recited in Claim 25, wherein the computer processor comprises a floating point logic unit (FPU), the FPU configured as a subunit of the computer processor.

33. (Previously Presented) A computer program product for conserving power in a computer processor, the computer program product having a medium with a computer program embodied thereon, the computer program comprising:

computer program code for reading a software-accessible control register; {page 1, para. 1, lines 1-2}

computer program code for determining an idle status of a subunit of the computer processor based on the control register; {page 1, para. 2, lines 4-10}

computer program code for providing a clock signal to the subunit based on the determined idle status; and {page 1, para. 2, lines 9-10; page 2}

computer program code for providing a power voltage to the subunit based on the determined idle status. {page 1, para. 2, lines 9-10; page 2}

34. (Previously Presented) The computer program product as recited in Claim 33, wherein the control register comprises a plurality of architected bits in a machine state register and at least one bit of the plurality of architected bits is associated with at least one subunit of the computer processor.

35. (Previously Presented) The computer program product as recited in Claim 34, wherein computer program code for determining the idle status comprises computer program code for reading the at least one bit associated with the at least one subunit of the computer processor.

36. (Previously Presented) The computer program product as recited in Claim 33, further comprising computer program code for setting one or more of a plurality of bits in the control register based on an idle status of a subunit of the computer processor.